On Query Answering in Description Logics with Number Restrictions on Transitive Roles*

Víctor Gutiérrez-Basulto[1], Yazmín Ibáñez-García[2], and Jean Christoph Jung[3]

[1] Cardiff University, UK (gutierrezbasultov@cardiff.ac.uk)
[2] TU Wien, Austria (ibanez@kr.tuwien.ac.at)
[3] Universität Bremen, Germany (jeanjung@informatik.uni-bremen.de)

Abstract. We study query answering in the description logic $\mathcal{SQ}$ supporting number restrictions on both transitive and non-transitive roles. Our main contributions are (i) a tree-like model property for $\mathcal{SQ}$ knowledge bases and, building upon this, (ii) an automata based decision procedure for answering two-way regular path queries, which gives a 3ExpTime upper bound.

1 Introduction

In the last years, several efforts have been put into the study of the query answering problem (QA) in description logics (DLs) featuring transitive roles (or generalisations thereof, such as regular expressions on roles) and number restrictions, see e.g., [10, 11, 9, 7, 8] and references therein. However, all these DLs heavily restrict the interaction between these two features, or altogether forbid number restrictions on transitive roles. Unfortunately, this comes as a shortcoming in crucial DL application areas like medicine and biology in which many terms, e.g., proteins, are defined and classified according to the number of components they contain or are part of (in a transitive sense) [27, 22, 24].

The lack of investigations of query answering in DLs of this kind is partly because (i) the interaction of these features often leads to undecidability of the standard reasoning tasks (e.g., satisfiability) - even in lightweight sub-Boolean DLs with unqualified number restrictions [17, 20, 15]; and (ii) for those DLs known to be decidable, such as $\mathcal{SQ}$ and $\mathcal{SOQ}$ [20, 18], only recently tight complexity bounds were obtained [15]. Moreover, even if these features (with restricted interaction) do not necessarily increase the complexity of QA, they do pose additional challenges for devising decision procedures [10, 11, 9] since they lead to the loss of properties, such as the tree model property, which make the design of algorithms for QA simpler. In fact, these difficulties are present already in DLs with transitivity, but without number restrictions [9]. Clearly, these issues are exacerbated if number restrictions are imposed on transitive roles.

The objective of this paper is to start the investigation of query answering in DLs supporting number restrictions on transitive roles. In particular, we look

* Gutiérrez-Basulto was funded by the EU's Horizon 2020 programme under the Marie Skłodowska-Curie grant No 663830

at the problem of answering regular path queries, which generalise standard query languages like positive existential queries, over $\mathcal{SQ}$ knowledge bases [16]. We first develop tree-like decompositions of $\mathcal{SQ}$-interpretations based on a novel unraveling that is specially tailored to handle the interaction of transitivity with number restrictions. Using these decompositions, we design an algorithm for the query answering problem using two-way alternating tree automata in the spirit of [10, 7, 8], resulting in a 3ExpTime upper bound (leaving an exponential gap).

Related Work. Schröder and Pattinson [23] investigate the DL $\mathcal{PHQ}$ supporting number restrictions on transitive parthood roles, which are, in contrast to $\mathcal{SQ}$, interpreted as trees: parthood-siblings cannot have a common part. They show that under this assumption decidability (for satisfiability) can be attained.

There has been some work on the extension of decidable first-order logic fragments, such as the guarded fragment, with transitivity and counting, see e.g., [25, 21]. Unfortunately, this case leads to undecidability unless the interaction is severely restricted [25]. Closer to DLs is the detailed investigation of modal logics with graded modalities carried out in [19]. Finally, in the context of existential rules, several efforts have been recently made to design languages with decidable QA supporting transitivity [12, 4, 1]. However, we are not aware of any attempts to additionally support number restrictions.

2 Preliminaries

Syntax. We introduce the DL $\mathcal{SQ}$, which extends the classical DL $\mathcal{ALC}$ with transitivity declarations on roles ($\mathcal{S}$) and qualified number restrictions ($\mathcal{Q}$). We consider a vocabulary consisting of countably infinite disjoint sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$, and assume that $\mathsf{N_R}$ is partitioned into two countably infinite sets of *non-transitive role names* $\mathsf{N_R^{nt}}$ and *transitive role names* $\mathsf{N_R^{t}}$. The syntax of *$\mathcal{SQ}$-concepts* C, D is given by the grammar rule $C, D ::= A \mid \neg C \mid C \sqcap D \mid (\leq n\ r\ C)$ where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and n is a number given in binary. We use $(\geq n\ r\ C)$ as an abbreviation for $\neg(\leq (n-1)\ r\ C)$, and other standard abbreviations like $\bot$, $\top$, $C \sqcup D$, $\exists r.C$, $\forall r.C$. Concepts of the form $(\leq n\ r\ C)$ and $(\geq n\ r\ C)$ are called *at most-restrictions* and *at least-restrictions*, respectively.

An *$\mathcal{SQ}$-TBox* $\mathcal{T}$ is a finite set of *concept inclusions* $C \sqsubseteq D$ where C, D are $\mathcal{SQ}$-concepts. An *ABox* is a finite set of *concept* and *role assertions* of the form $A(a)$, $r(a, b)$ where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$ and $\{a, b\} \subseteq \mathsf{N_I}$; $\mathsf{ind}(\mathcal{A})$ denotes the set of individual names occurring in $\mathcal{A}$. A *knowledge base (KB)* $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$.

Semantics. As usual, the semantics is defined in terms of interpretations. An *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty *domain* $\Delta^\mathcal{I}$ and an *interpretation function* $\cdot^\mathcal{I}$ mapping concept names to subsets of the domain and role names to binary relations over the domain such that transitive role names are mapped to transitive relations. We define $C^\mathcal{I}$ for complex concepts C by interpreting $\neg$ and $\sqcap$ as usual and $(\leq n\ r\ D)^\mathcal{I}$ by taking

$$(\leq n\ r\ D)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid \ |\{e \in C^\mathcal{I} \mid (d, e) \in r^\mathcal{I}\}| \leq n\}.$$

For ABoxes $\mathcal{A}$ we adopt the *standard name assumption (SNA)*, that is, $a^{\mathcal{I}} = a$, for all $a \in \mathsf{ind}(\mathcal{A})$, but we conjecture that our results hold without the unique name assumption. The satisfaction relation $\models$ is defined in the standard way:

$$\mathcal{I} \models C \sqsubseteq D \text{ iff } C^{\mathcal{I}} \subseteq D^{\mathcal{I}}; \quad \mathcal{I} \models A(a) \text{ iff } a \in A^{\mathcal{I}}; \quad \mathcal{I} \models r(a,b) \text{ iff } (a,b) \in r^{\mathcal{I}}.$$

An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, denoted $\mathcal{I} \models \mathcal{T}$, if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{T}$; it is a model of an ABox $\mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$, if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{A}$; it is a model of a KB $\mathcal{K}$ if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. A KB is *satisfiable* if it has a model.

Query Language. As query language, we consider *regular path queries*, supporting regular expressions over roles. Recall that a *regular expression $\mathcal{E}$ over an alphabet Σ* is given by the grammar $\mathcal{E} ::= \varepsilon \mid \sigma \mid \mathcal{E}\cdot\mathcal{E} \mid \mathcal{E} \cup \mathcal{E} \mid \mathcal{E}^*$, where $\sigma \in \Sigma$ and ε denotes the *empty word*. We denote with $L(\mathcal{E})$ the language defined by $\mathcal{E}$.

We use $\mathsf{N}_{\mathsf{R}}^{\pm}$ to refer to $\mathsf{N}_{\mathsf{R}} \cup \{r^- \mid r \in \mathsf{N}_{\mathsf{R}}\}$ with $(r^-)^{\mathcal{I}}$ defined as $\{(d,e) \mid (e,d) \in r^{\mathcal{I}}\}$, and identify r^- with $s \in \mathsf{N}_{\mathsf{R}}$ if $r = s^-$. A *positive 2-way regular path query (P2RPQ)* is a formula of the form $q(\boldsymbol{x}) = \exists \boldsymbol{y}.\varphi(\boldsymbol{x},\boldsymbol{y})$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are tuples of variables and φ is constructed using $\wedge$ and $\vee$ of atoms of the form $A(t)$ or $\mathcal{E}(t,t')$ where $A \in \mathsf{N}_{\mathsf{C}}$, $\mathcal{E}$ is a regular expression over $\mathcal{S} ::= \mathsf{N}_{\mathsf{R}}^{\pm} \cup \{A? \mid A \in \mathsf{N}_{\mathsf{C}}\}$, and t,t' are *terms*, i.e., individual names or variables from $\boldsymbol{x},\boldsymbol{y}$. We define as usual when a possible answer tuple $\boldsymbol{a} \in \mathsf{ind}(\mathcal{A})$ is a *certain answer of q over $\mathcal{K}$* and write $\mathcal{K} \models q(\boldsymbol{a})$ in case it is [8, 6].

Reasoning Problem. We study the *certain answers problem:* Given a KB $\mathcal{K}$, a query $q(\boldsymbol{x})$ and a tuple of individuals $\boldsymbol{a}$, determine whether $\mathcal{K} \models q(\boldsymbol{a})$. Without loss of generality, we consider Boolean queries.

3 Decomposing $\mathcal{SQ}$-Interpretations

Existing algorithms for QA in expressive DLs, e.g., $\mathcal{SHIQ}$ (without number restrictions on transitive roles), exploit the fact that for answering queries it suffices to consider *canonical models* that are forest-like, roughly consisting of an interpretation of the ABox and a collection of tree-interpretations whose roots are elements of the ABox. However, for $\mathcal{SQ}$ this tree-model property is lost:

Example 1. Let $\mathcal{T} = \{A \sqsubseteq (\leq 1\ r\ C) \sqcap \exists r.B \sqcap \exists r.\neg B, \top \sqsubseteq \exists r.C\}$ with $r \in \mathsf{N}_{\mathsf{R}}^{t}$. The number restrictions in $\mathcal{T}$ force that every model of $\mathcal{T}$ satisfying A contains the structure in Fig. 1(a). Moreover, in $\mathcal{SQ}$ strongly connected components can be enforced. Let $\mathcal{T}' = \{A \sqsubseteq (= 3\ r\ B), B \sqsubseteq (= 3\ r\ B), A \sqsubseteq \neg B\}$ with $r \in \mathsf{N}_{\mathsf{R}}^{t}$. Then, in every model of $\mathcal{T}'$, an element satisfying A roots the structure depicted in Fig. 1(b), where the elements satisfying B form a strongly connected component.

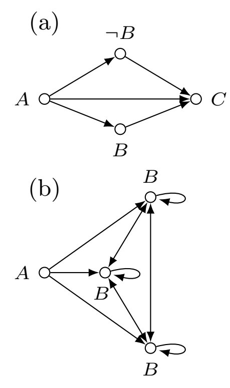

Fig. 1.

Nevertheless, we will define *tree-like* canonical models for $\mathcal{SQ}$ that suffice for query answering. We start with introducing a basic form of tree decompositions of $\mathcal{SQ}$-interpretations.

A *tree* is a connected, acyclic graph (T, E) with a distinguished root, which we usually denote with ε. We usually write only T instead of (T, E), thus leaving E implicit. Fix some interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. A *bag* M is a set of assertions of the form $A(d), r(d, e)$ with $d, e \in \Delta^{\mathcal{I}}$. We denote with $\mathsf{dom}(M)$ the set of domain elements appearing in M. Given a set $\Lambda \subseteq \Delta^{\mathcal{I}}$, we denote with $\mathsf{bag}_{\mathcal{I}}(\Lambda)$ the set

$$\mathsf{bag}_{\mathcal{I}}(\Lambda) = \{A(d) \mid d \in \Lambda, d \in A^{\mathcal{I}}\} \cup \{r(d, e) \mid d, e \in \Lambda, (d, e) \in r^{\mathcal{I}}\}.$$

Definition 2. *A* tree decomposition of an interpretation $\mathcal{I}$ *is a tuple* (T, bg) *where* T *is a tree and* bg *assigns a bag to every node* w *in* T *such that:*

(i) $\mathsf{bg}(w) = \mathsf{bag}_{\mathcal{I}}(\mathsf{dom}(\mathsf{bg}(w)))$*;*
(ii) $\Delta^{\mathcal{I}} = \bigcup_{w \in T} \mathsf{dom}(\mathsf{bg}(w))$*;*
(iii) $r^{\mathcal{I}} = \chi_r$ *for non-transitive* r *and* $r^{\mathcal{I}} = \chi_r^+$ *for transitive* r*, where*

$$\chi_r = \{(d, e) \mid r(d, e) \in \mathsf{bg}(w), w \in T\};$$

(iv) for all $d \in \Delta^{\mathcal{I}}$*, the set* $\{w \in T \mid d \in \mathsf{dom}(\mathsf{bag}(w))\}$ *is connected in* T*.*

Definition 2 provides a variant of tree decompositions of interpretations with transitive relations. This formalisation does not yet enable tree automata to count over transitive roles (with a small number of states) since assertions $r(d, e)$ can appear far away from each other in the decomposition. To address this, we introduce *canonical tree decompositions* which extend tree decompositions with a third component rl which assigns a role name to every non-root node of T and $\bot$ to the root ε. Intuitively, a node $w \in T$ labeled with $r = \mathsf{rl}(w)$ will be responsible for capturing r-successors of some element(s) in the previous bag.

Fix a triple $(T, \mathsf{bg}, \mathsf{rl})$ such that (T, bg) is a tree decomposition of $\mathcal{I}$. By Item *(iv)* of Definition 2, for each $d \in \Delta^{\mathcal{I}}$, there is a unique node $w \in T$ such that all occurrences of d are in or below w in T. In this case, we say that d is *fresh* in w, and we denote with $F(w)$ the set of all fresh elements in w. We will also need a relativised version of freshness which takes into account the role associated to the predecessor bag. In particular, for each transitive role r and each $w \in T$ with $\mathsf{rl}(w) \in \{r, \bot\}$, we define a set $F_r(w)$ by taking:

- $F_r(w) = F(w)$ if the predecessor $\hat{w}$ of w satisfies $\mathsf{rl}(\hat{w}) \in \{r, \bot\}$;
- $F_r(w) = \mathsf{dom}(\mathsf{bg}(w))$ otherwise.

Intuitively, $F_r(w)$ contains all elements that are eligible as origins for r-successors.

For a transitive role r and a bag M, we call $\emptyset \subsetneq \mathbf{a} \subseteq \mathsf{dom}(M)$ an *r-cluster in M* if *(i)* $r(a, b) \in M$ for all $a \neq b \in \mathbf{a}$, and *(ii)* for all $a \in \mathbf{a}$, $b \in \mathsf{dom}(M)$ with $r(a, b), r(b, a) \in M$, we have $b \in \mathbf{a}$. An r-cluster $\mathbf{a}$ in M is an *r-root cluster in M* if $r(d, e) \in M$ for all $d \in \mathbf{a}$ and $e \in \mathsf{dom}(M) \setminus \mathbf{a}$.

Definition 3. *A triple* $\mathfrak{T} = (T, \mathsf{bg}, \mathsf{rl})$ *is a* canonical tree decomposition of $\mathcal{I}$ *if* (T, bg) *is a tree decomposition of* $\mathcal{I}$ *and the following conditions are satisfied for every* $w \in T$ *with* $M = \mathsf{bg}(w)$ *and* $r = \mathsf{rl}(w)$ *and every successor* w' *of* w *with* $M' = \mathsf{bg}(w')$ *and* $r' = \mathsf{rl}(w')$*:*

(C1) if $r' \in \mathsf{N}_\mathsf{R}^{nt}$*, then* $\mathsf{dom}(M') = \{d, e\}$*, for some* $d \in F(w)$*,* $e \in F(w')$*, and* $r'(d, e)$ *is the only role assertion in* M'*;*

(C2) *if* $r' \in \mathsf{N}_\mathsf{R}^t$ *and* $r \notin \{\bot, r'\}$*, then there are* $d \in F(w)$ *and an* r*-root cluster* $\mathbf{a}$ *in* M' *such that* $\mathsf{dom}(M) \cap \mathsf{dom}(M') = \{d\}$ *and* $d \in \mathbf{a}$*; moreover, there is no successor* v' *of* w *different from* w' *satisfying this for* d *and* $\mathsf{rl}(v') = r'$*;*

(C3) *if* $r' \in \mathsf{N}_\mathsf{R}^t$ *and* $r \in \{\bot, r'\}$*, then there is an* r'*-cluster* $\mathbf{a}$ *in* M *with:*
- *(a)* $\mathbf{a} \subseteq F_{r'}(w)$*;*
- *(b)* $\mathbf{a}$ *is an* r'*-root cluster in* M'*;*
- *(c)* *for all* $d \in \mathbf{a}$ *and* $r'(d, e) \in M$*, we have* $e \in \mathsf{dom}(M')$*; and*
- *(d)* *for all* $r'(d, e) \in M'$*,* $d \in \mathbf{a} \cup F(w')$ *or* $e \notin F(w')$*.*

Definition 3 imposes restrictions on the structural relation between neighbouring bags. Note that Condition (C1) is also satisfied by standard unravelings over non-transitive roles [2]. Condition (C2) reflects that neighbouring bags associated with different role names do only interact via single domain elements; this conforms with viewing $\mathcal{SQ}$ as a fusion logic [3]. Most interestingly, Condition (C3) plays the role of (C1), but for transitive roles. It is important to note that (C3) is based on r-clusters now since they can be enforced, see Example 1 above. Item *(a)* restricts for which elements $\mathbf{a}$ we can have successor bags; Item *(b)* requires that $\mathbf{a}$ is a root cluster in the successor bag M'; Item *(c)* states that everything which was reachable from $\mathbf{a}$ via r' in M should be also included in M'; finally, Item *(d)* requires that there are no connections $r(d, e) \in M'$ between elements d from M and fresh elements e from M'.

As a consequence of Definition 3, we can address r-successors of elements in a canonical tree decomposition $\mathfrak{T} = (T, \mathsf{bg}, \mathsf{rl})$ of $\mathcal{I}$. For a non-transitive role r, Condition (C1) ensures that r-successors e of d are contained only in successor nodes of the (unique) node where d is fresh. For a transitive role r, note first that $(d, e) \in r^\mathcal{I}$ iff there is a sequence $d_0, w_0, d_1, \ldots, w_{n-1}, d_n$ with $d_i \in \Delta^\mathcal{I}$ and $w_i \in T$ such that $d = d_0$, $e = d_n$, and $r(d_i, d_{i+1}) \in \mathsf{bg}(w_i)$, for all $0 \leq i < n$; we call such a sequence an *r-path from d to e in $\mathfrak{T}$*. We call an r-path $d_0, w_0, d_1, \ldots, w_{n-1}, d_n$ *downward* if w_i is a successor of w_{i-1} and d_i is contained in an r-root cluster of w_i, for all $0 < i < n$. An r-path in $\mathfrak{T}$ is *canonical* if **P1**: it is downward; or **P2**: $d_0 \in F_r(w_0)$, $d_1 \notin F_r(w_0)$, and, if $n > 1$, then $d_1, w_1, \ldots, d_n$ is a downward path in $\mathfrak{T}$ and the predecessor $\hat{w}$ of w_1 is an ancestor of w_0 and satisfies $d_1 \in F_r(\hat{w})$. Two r-paths $d_0, w_0, d_1, \ldots, w_{n-1}, d_n$ and $e_0, w'_0, e_1, \ldots, w'_{m-1}, e_m$ from d to e are *equivalent* if $n = m$, $w_i = w'_i$, for $0 \leq i < n$, and d_i and e_i are in the same r-cluster in $\mathsf{bg}(w_i)$, for every $1 \leq i < n$.

Lemma 4 establishes the basis for uniquely identifying transitive r-successors in a canonical tree decomposition which is essential for the design of automata.

Lemma 4. *Let $\mathfrak{T}$ be a canonical decomposition of $\mathcal{I}$, r transitive, and $(d, e) \in r^\mathcal{I}$. Then there is a unique canonical r-path up to equivalence from d to e in $\mathfrak{T}$.*

3.1 Unraveling into Canonical Decompositions

We give now the main technical contribution of our paper: an unraveling operation into canonical decompositions of small width, and consequently a tree-like model property for $\mathcal{SQ}$-interpretations. A canonical tree decomposition $(T, \mathsf{bg}, \mathsf{rl})$ has *width* $k - 1$ if k is the maximum size of $\mathsf{dom}(\mathsf{bg}(w))$, where w ranges over T; its *outdegree* is the outdegree of the underlying tree T.

Theorem 5. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{SQ}$ KB and $\mathcal{I} \models \mathcal{K}$. Then, there is an interpretation $\mathcal{J}$ and a canonical tree decomposition $(T, \mathsf{bg}, \mathsf{rl})$ of $\mathcal{J}$ such that:*

(1) $\mathcal{A} \subseteq \mathsf{bg}(\varepsilon)$;
(2) $\mathcal{J} \models \mathcal{K}$;
(3) there is a homomorphism from $\mathcal{J}$ to $\mathcal{I}$;
(4) width and outdegree of $(T, \mathsf{bg}, \mathsf{rl})$ are bounded by $O(|\mathcal{A}| \cdot 2^{\mathsf{poly}(|\mathcal{T}|)})$.

We outline the proof of Theorem 5. As a first step, we show that wlog. we can assume that $\mathcal{I}$ has a restricted outdegree and width, as defined below. This will be used later on to ensure the satisfaction of Condition (4) above. Given $d \in \Delta^{\mathcal{I}}$ and a transitive role r, the *r-cluster of d in $\mathcal{I}$*, denoted by $Q_{\mathcal{I},r}(d)$, is the set of all elements $e \in \Delta^{\mathcal{I}}$ such that both $(d, e) \in r^{\mathcal{I}}$ and $(e, d) \in r^{\mathcal{I}}$. The *width of $\mathcal{I}$* is the minimum k such that $|Q_{\mathcal{I},r}(d)| \leq k$ for all $d \in \Delta^{\mathcal{I}}$, $r \in \mathsf{N}_{\mathsf{R}}^t$. Moreover, for a transitive role r, we say that e is a *direct r-successor of d* if $(d, e) \in r^{\mathcal{I}}$ but $e \notin Q_{\mathcal{I},r}(d)$, and for each f with $(d, f), (f, e) \in r^{\mathcal{I}}$, we have $f \in Q_{\mathcal{I},r}(d)$ or $f \in Q_{\mathcal{I},r}(e)$; if r is non-transitive, then e is a *direct r-successor of d* if $(d, e) \in r^{\mathcal{I}}$. The *breadth of $\mathcal{I}$* is the maximum k such that there are $d, d_1, \dots, d_k$ and a role name r, all d_i are direct r-successors of d, and

– if r is non-transitive, then $d_i \neq d_j$ for all $i \neq j$;
– if r is transitive, then $Q_{\mathcal{I},r}(d_i) \neq Q_{\mathcal{I},r}(d_j)$, for all $i \neq j$.

We can assume that width and breath of $\mathcal{I}$ are within the following boundaries.

Lemma 6 (adapting [19, 15]). *For each $\mathcal{I} \models \mathcal{K}$, there is a sub-interpretation $\mathcal{I}'$ of $\mathcal{I}$ with $\mathcal{I}' \models \mathcal{K}$ and width and breadth of $\mathcal{I}'$ are bounded by $O(|\mathcal{A}| + 2^{\mathsf{poly}(|\mathcal{T}|)})$.*

We need to introduce one more notion for dealing with at-most restrictions over transitive roles. Let $\mathsf{cl}(\mathcal{T})$ be the set of all subconcepts occurring in $\mathcal{T}$, closed under single negation. For each transitive role r, define a binary relation $\rightsquigarrow_{\mathcal{I},r}$ on $\Delta^{\mathcal{I}}$, by taking $d \rightsquigarrow_{\mathcal{I},r} e$ if there is some $(\leq n\ r\ C) \in \mathsf{cl}(\mathcal{T})$ such that $d \in (\leq n\ r\ C)^{\mathcal{I}}$, $e \in C^{\mathcal{I}}$, and $(d, e) \in r^{\mathcal{I}}$. Based on the transitive, reflexive closure $\rightsquigarrow^*_{\mathcal{I},r}$ of $\rightsquigarrow_{\mathcal{I},r}$, we define, for every $d \in \Delta^{\mathcal{I}}$, the set $\mathsf{Wit}_{\mathcal{I},r}(d)$ of *witnesses for d* by taking

$$\mathsf{Wit}_{\mathcal{I},r}(d) = \bigcup\nolimits_{e \mid d \rightsquigarrow^*_{\mathcal{I},r} e} Q_{\mathcal{I},r}(e).$$

Intuitively, $\mathsf{Wit}_{\mathcal{I},r}(d)$ contains all witnesses of at-most restrictions of some element d, and due to using $\rightsquigarrow^*_{\mathcal{I},r}$, also the witnesses of at-most restrictions of those witnesses and so on. It is important to note that the size of $\mathsf{Wit}_{\mathcal{I},r}(\mathcal{T})$ is bounded exponentially in $\mathcal{T}$ (and linearly in $\mathcal{A}$), see appendix.

We describe now the construction of the interpretation $\mathcal{J}$ and its tree decomposition via a possibly infinite unraveling process. Elements of $\Delta^{\mathcal{J}}$ will be either of the form a with $a \in \mathsf{ind}(\mathcal{A})$ or of the form d_x with $d \in \Delta^{\mathcal{I}}$ and some index x. We usually use δ to refer to domain elements in $\mathcal{J}$ (in either form), and define a function $\tau : \Delta^{\mathcal{J}} \to \Delta^{\mathcal{I}}$ by setting $\tau(\delta) = \delta$, for all $\delta \in \mathsf{ind}(\mathcal{A})$, and $\tau(\delta) = d$, for all δ of the form d_x in $\Delta^{\mathcal{J}}$.

To start the construction of $\mathcal{J}$ and $(T, \mathsf{bg}, \mathsf{rl})$, we set $\mathcal{J} = \mathcal{I}|_{\mathsf{ind}(\mathcal{A})}$ and, for every transitive role r, define two sets Δ_r, Δ'_r by taking

$$\Delta_r = \{d_r \mid d \in \bigcup_{a\in\mathsf{ind}(\mathcal{A})} \mathsf{Wit}_{\mathcal{I},r}(a) \setminus \mathsf{ind}(\mathcal{A})\} \quad \text{and} \quad \Delta'_r = \Delta_r \cup \mathsf{ind}(\mathcal{A}).$$

Then extend $\mathcal{J}$ by adding, for each transitive r, Δ_r to the domain and extending the interpretation of concept and role names such that, for all $\delta, \delta' \in \Delta'_r$, we have

$$\delta \in A^{\mathcal{J}} \Leftrightarrow \tau(\delta) \in A^{\mathcal{I}}, \text{ for all } A \in \mathsf{N_C}, \text{ and } (\delta,\delta') \in r^{\mathcal{J}} \Leftrightarrow (\tau(\delta),\tau(\delta')) \in r^{\mathcal{I}}. \quad (\dagger)$$

Now, initialise $(T, \mathsf{bg}, \mathsf{rl})$ with $T = \{\varepsilon\}$, $\mathsf{bg}(\varepsilon) = \mathsf{bag}_{\mathcal{J}}(\Delta^{\mathcal{J}})$, and $\mathsf{rl}(\varepsilon) = \bot$. Intuitively, this first step ensures that all witnesses of ABox individuals appear in the first bag. This finishes the initialisation phase.

Next, extend $\mathcal{J}$ and $(T, \mathsf{bg}, \mathsf{rl})$ by applying the following rules exhaustively and in a fair way:

R₁ Let r be non-transitive, $w \in T$, $\delta \in F(w)$, and d a direct r-successor of $\tau(\delta)$ in $\mathcal{I}$ with $\{\delta, d\} \not\subseteq \mathsf{ind}(\mathcal{A})$. Then, add a fresh successor v of w to T, add a fresh element d_v to $\Delta^{\mathcal{J}}$, extend $\mathcal{J}$ by adding $(\delta, d_v) \in r^{\mathcal{J}}$ and $d_v \in A^{\mathcal{J}}$ iff $d \in A^{\mathcal{I}}$, for all $A \in \mathsf{N_C}$, and set $\mathsf{bg}(v) = \mathsf{bag}_{\mathcal{J}}(\{\delta, d_v\})$ and $\mathsf{rl}(v) = r$.

R₂ Let r be transitive, $w \in T$, and $\delta \in F(w)$ such that:
(a) $w = \varepsilon$ and $\delta \in \Delta_s$, $s \neq r$ (Δ_s defined in the initialisation phase), or
(b) $w \neq \varepsilon$ and $\mathsf{rl}(w) \neq r$.
Then add a fresh successor v of w to T, and define

$$\Delta = \{e_v \mid e \in \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta)) \setminus \{\tau(\delta)\}\} \quad \text{and} \quad \Delta' = \Delta \cup \{\delta\}.$$

Extend the domain of $\mathcal{J}$ with Δ and the interpretation of concept and role names such that $(\dagger)$ is satisfied for all $\delta, \delta' \in \Delta'$. Finally, set $\mathsf{bg}(v) = \mathsf{bag}_{\mathcal{J}}(\Delta')$ and $\mathsf{rl}(v) = r$.

R₃ Let r be transitive, $w \in T$, $\mathbf{a} \subseteq F_r(w)$ an r-cluster in $\mathsf{bg}(w)$ such that:
(a) $w = \varepsilon$ and $\mathbf{a} \subseteq \Delta'_r$, or
(b) $w \neq \varepsilon$ and $\mathsf{rl}(w) = r$.
If there is a direct r-successor e of $\tau(\delta)$ in $\mathcal{I}$ for some $\delta \in \mathbf{a}$ such that $(\delta, \delta') \notin r^{\mathcal{J}}$ for any δ' with $\tau(\delta') = e$, then add a fresh successor v of w to T, and define

$$\Delta = \{f_v \mid f \in \mathsf{Wit}_{\mathcal{I},r}(e) \setminus \mathsf{Wit}_{\mathcal{I},r}(\tau(\delta))\} \quad \text{and}$$
$$\Delta' = \Delta \cup \mathbf{a} \cup \{\delta'' \mid r(\delta', \delta'') \in \mathsf{bg}(w) \text{ for some } \delta' \in \mathbf{a}\}.$$

Then extend the domain of $\mathcal{J}$ with Δ and the interpretation of concept names such that $(\dagger)$ is satisfied for all pairs δ, δ' with $\delta \in \mathbf{a} \cup \Delta$ and $\delta' \in \Delta'$. Finally, set $\mathsf{bg}(v) = \mathsf{bag}_{\mathcal{J}}(\Delta')$ and $\mathsf{rl}(v) = r$.

Rules **R₁**–**R₃** are, respectively, in one-to-one correspondence with Conditions (C1)-(C3) in Definition 3. In particular, **R₁** implements the well-known unraveling procedure for non-transitive roles. **R₂** is used to change the 'role component' for transitive roles by creating a bag which contains all witnesses of the chosen element δ. Finally, **R₃** describes how to unravel direct r-successors in case of transitive roles r. In the definition of Δ it is taken care that witnesses which are 'inherited' from predecessors are not introduced again, in order to preserve at-most restrictions.

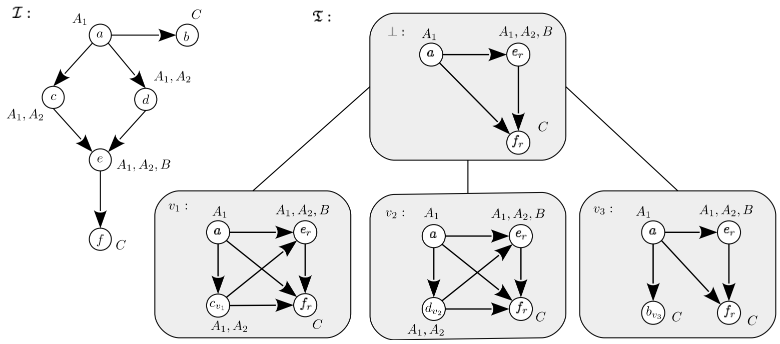

Fig. 2.

Example 7. Let $\mathcal{K} = (\{A_1 \sqsubseteq (\leq 1\ r\ B), A_2 \sqsubseteq (\leq 1\ r\ C)\}, \{A_1(a)\})$ with $r \in \mathsf{N}_\mathsf{R}^t$. Fig. 2 shows a model $\mathcal{I}$ of $\mathcal{T}$ and a canonical decomposition $\mathfrak{T}$ of its unraveling (transitivity connections are omitted). In the initialisation phase, the $\bot$-bag is constructed starting from individual a. Since $a \rightsquigarrow_{\mathcal{I},r} e$ and $e \rightsquigarrow_{\mathcal{I},r} f$, we have $\mathsf{Wit}_{\mathcal{I},r}(a) = \{e, f\}$, thus e_r and f_r are added in this phase.

It is verified in the appendix that $(T, \mathsf{bg}, \mathsf{rl})$ and $\mathcal{J}$ satisfy the conditions from Theorem 5. Theorem 5 yields a tree-like model property for $\mathcal{SQ}$-knowledge bases, which is interesting on its own since existing decidability results (for satisfiability) [20, 15] are based on the finite model property.

4 Automata-Based Approach to Query Answering

In this section, we devise an automata-based decision procedure for query answering in $\mathcal{SQ}$. By Theorem 5, if $\mathcal{K} \not\models q$, there is an interpretation of small width and outdegree witnessing this. The idea is now to design two automata $\mathfrak{A}_\mathcal{K}$ and $\mathfrak{A}_q$ working over tree decompositions which accept precisely the models (of a fixed width) of the KB $\mathcal{K}$ and the query q, respectively. Query answering is then reduced to the question if some tree is accepted by $\mathfrak{A}_\mathcal{K}$, but not by $\mathfrak{A}_q$ [8].

Trees are represented as prefix-closed subsets $T \subseteq (\mathbb{N} \setminus \{0\})^*$ such that additionally, $wc \in T$ implies $w(c-1) \in T$ for all $c > 1$. A tree is *k-ary* if each node has *exactly* k successors. As a convention, we set $w \cdot 0 = w$ and $wc \cdot (-1) = w$, leave $\varepsilon \cdot (-1)$ undefined, and for any $k \in \mathbb{N}$, set $[k] = \{-1, 0, \ldots, k\}$. Let Σ be a finite alphabet. A *Σ-labeled tree* is a pair (T, τ) with T a tree and $\tau : T \to \Sigma$ assigns a letter from Σ to each node. An *alternating 2-way tree automaton (2ATA)* over Σ-labeled k-ary trees is a tuple $\mathfrak{A} = (Q, \Sigma, q_0, \delta, F)$ where Q is a finite set of *states*, $q_0 \in Q$ is an *initial state*, δ is the *transition function*, and F is the *(parity) acceptance condition* [26]. The transition function maps a state q and an input letter $a \in \Sigma$ to a positive Boolean formula over the constants true and false, and variables from $[k] \times Q$. The semantics is given in terms of *runs*, see the appendix. As usual, $L(\mathfrak{A})$ denotes the set of trees accepted by $\mathfrak{A}$. Emptiness of $L(\mathfrak{A})$ can be checked in exponential time in the number of states of $\mathfrak{A}$ [26].

In principle, tree decompositions $\mathfrak{T}$ can be represented as labeled trees, where each node label consists of a bag and a role name (or $\bot$). However, 2ATAs cannot run over such labeled trees because the domain underlying the bags is potentially infinite. Exploiting the bounded width, we encode the infinite domain with finitely many elements in the following well-known way [14, 5]. Let $\mathcal{K}$ be an $\mathcal{SQ}$ KB, let K be the bound on the width obtained in Theorem 5, and choose a set of elements Δ of size $2K$. We define the input alphabet Σ as the set of all pairs $\langle M, x\rangle$ such that M is a bag that uses only constants from Δ, $|\mathsf{dom}(M)| \leq K$, and x is a role appearing in $\mathcal{K}$ or $\bot$. A Σ-labeled tree (T, τ) represents a tree decomposition (and thus an interpretation) as follows. Each domain element $d \in \Delta$ induces an equivalence relation $\sim_d$ on T by taking $v \sim_d w$ iff d appears in all bags on the path from v to w. Domain elements in the represented interpretation are then all equivalence classes obtained in this way. Moreover, for all $w \in T$, $\tau(w)$ represents the following bag:

$$\mathsf{bg}(w) = \{A([w]_{\sim_d}) \mid A(d) \in \tau(w))\} \cup \{r([w]_{\sim_d}, [w]_{\sim_e}) \mid r(d,e) \in \tau(w)\}.$$

We denote the interpretation associated with a Σ-labeled tree (T, τ) with $\mathcal{I}_{T,\tau}$. Moreover, we consider only k-ary trees where k is the bound on the outdegree given by Theorem 5. Since tree decompositions are not necessarily uniformly branching, we include an auxiliary symbol $\bullet$ to refer to non-existing branches.

Lemma 8. *There are 2ATAs $\mathfrak{A}_1, \mathfrak{A}_2, \mathfrak{A}_3$ of size $O(|\mathcal{A}| \cdot 2^{\mathsf{poly}(|\mathcal{T}|)})$ such that: $(T, \tau) \in L(\mathfrak{A}_1)$ iff (T, τ) is the canonical decomposition of some interpretation; $(T, \tau) \in L(\mathfrak{A}_2)$ iff $\mathcal{I}_{T,\tau} \models \mathcal{A}$, and $(T, \tau) \in L(\mathfrak{A}_3)$ iff $\mathcal{I}_{T,\tau} \models \mathcal{T}$.*

The mentioned automaton $\mathfrak{A}_{\mathcal{K}}$ is obtained as the conjunction of $\mathfrak{A}_1$, $\mathfrak{A}_2$, and $\mathfrak{A}_3$. Note that $\mathfrak{A}_{\mathcal{K}}$ can be used to decide KB satisfiability in double exponential time, thus not optimal [15]. We concentrate here on the most interesting 2ATA $\mathfrak{A}_3$. Denote with $\mathsf{nnf}(C)$ the *negation normal form* of a concept C, and define $\mathsf{nnf}(\mathcal{T}) = \{\mathsf{nnf}(C) \mid C \in \mathsf{cl}(\mathcal{T})\}$. Moreover, let $\mathsf{Rol}(\mathcal{K})$ be the set of role names appearing in $\mathcal{K}$. Then, define $\mathfrak{A}_3 = (Q_3, \Sigma, q_0, \delta_3, F_3)$; start by including in Q_3

$$\{q_0\} \cup Q^{nt} \cup Q^t \cup \{F_{x,d}, F'_{x,d}, \overline{F}_{x,d}, \overline{F}'_{x,d} \mid d \in \Delta, x \in \{\bot\} \cup \mathsf{Rol}(\mathcal{K})\} \cup$$
$$\{q_d, q_{C,d} \mid C \in \mathsf{nnf}(\mathcal{T}), d \in \Delta\} \cup \{q^*_{C,d}, q'_{C,d} \mid C = (\sim\ n\, r\, D) \in \mathsf{nnf}(\mathcal{T}), d \in \Delta\}$$

where Q^t and Q^{nt} are the states that are used after entering states $q^*_{(\sim n\, r\, D),d}$ for transitive and non-transitive roles, respectively. Then, we define the transition function for all states except states of the form $q^*_{(\sim n\, r\, D),d}$:

$$\delta_3(q_0, \langle M, x\rangle) = \bigwedge_{i \in [k]} (i, q_0) \wedge \bigwedge_{d \in \mathsf{dom}(M)} \bigwedge_{C \sqsubseteq D \in \mathcal{T}} \big((0, q_{\mathsf{nnf}(\neg C),d}) \vee (0, q_{D,d})\big)$$
$$\delta_3(q_0, \bullet) = \mathsf{true}$$
$$\delta_3(q_{A,d}, \langle M, x\rangle) = \text{if } A(d) \in M, \text{ then } \mathsf{true} \text{ else } \mathsf{false}$$
$$\delta_3(q_{\neg A,d}, \langle M, x\rangle) = \text{if } A(d) \notin M, \text{ then } \mathsf{true} \text{ else } \mathsf{false}$$
$$\delta_3(q_{C_1 \sqcup C_2,d}, \langle M, x\rangle) = (0, q_{C_1,d}) \vee (0, q_{C_2,d})$$

$$\delta_3(q_{C_1\sqcap C_2,d},\langle M,x\rangle) = (0,q_{C_1,d})\wedge(0,q_{C_2,d})$$

$$\delta_3(q_{(\sim n\,r\,D),d},\langle M,x\rangle) = \big((0,F_{x,d})\wedge(0,q^*_{(\sim n\,r\,D),d})\big)\vee\bigvee_{i\in[k]}(i,q_{(\sim n\,r\,D),d})\wedge(i,q_d)$$

$$\delta_3(q_d,\langle M,x\rangle) = \text{if } d\in\mathsf{dom}(M)\text{, then }\ \mathsf{true}\ \text{ else }\ \mathsf{false}$$

$$\delta_3(F_{\bot,d},\langle M,x\rangle) = \mathsf{true}$$

$$\delta_3(F_{r,d},\langle M,x\rangle) = \begin{cases}(-1,F'_{r,d}) & \text{if } r\in\mathsf{N}_\mathsf{R}^{nt}\text{ or }(r\in\mathsf{N}_\mathsf{R}^{t}\text{ and }x=r)\\ \mathsf{false} & \text{otherwise}\end{cases}$$

$$\delta_3(F'_{r,d},\langle M,x\rangle) = \begin{cases}\mathsf{true} & d\notin\mathsf{dom}(M)\text{ or }(r\in\mathsf{N}_\mathsf{R}^{t}\text{ and }x\notin\{\bot,r\})\\ \mathsf{false} & \text{otherwise}\end{cases}$$

Intuitively, q_0 is used to verify that the TBox is *globally* satisfied. A state $q_{C,d}$ assigned to a node w is used as an obligation to verify that the element d satisfies the concept C. This can be done locally for Boolean concept constructors $\sqcap,\sqcup,\neg$, as implemented in the transitions above. For concepts of the form $(\sim n\ r\ D)$, we have to be more careful: the automaton has to move to the unique node w where $d\in F_r(w)$, identified using states $F_{r,d}$ and q_d (and the acceptance condition).

The transitions for number restrictions on non-transitive roles are defered to the appendix. For transitive roles, Lemma 4 provides the following observation: For counting the r-successors satisfying D of some $d\in\mathsf{dom}(\mathsf{bg}(w))$, it suffices to look at three "locations" in the tree decomposition: in the bag at w itself, along canonical paths satisfying **P1**, and along canonical paths satisfying **P2**. We next implement this strategy for at-least restrictions. In the following transitions, we assume that $\mathbf{a}_1,\ldots,\mathbf{a}_\ell$ are all r-clusters in M, and that $a_1,\ldots,a_\ell$ are representatives of each cluster. A partition $n_1+\ldots+n_\ell=n$ *respects M relative to d* if $n_i=0$ whenever $r(d,a_i)\notin M$; it *d-respects M relative to d* if $n_i=0$ whenever $r(d,a_i)\notin M$ or $d\in\mathbf{a}_i$. Moreover, we define $M_r(d)=\{e\mid r(d,e),r(e,d)\in M\}$, and define transitions for (the complement of $F_{x,d}$) $\overline{F}_{x,d}$ similar to $F_{x,d}$.

$$\delta_3(q^*_{(\geq n\,r\,D),d},\langle M,x\rangle) = \bigvee_{\substack{n_1+\ldots+n_\ell=n\\ \text{respects } M \text{ rel. to } d}}\ \bigwedge_{n_i\neq 0}(0,q^0_{(\geq n_i\,r\,D),a_i})\vee(0,q^1_{(\geq n_i\,r\,D),a_i})$$

$$\delta_3(q^0_{(\geq n\,r\,D),d},\langle M,x\rangle) = (0,F_{r,d})\wedge(0,q^\downarrow_{(\geq n\,r\,D),d})$$

$$\delta_3(q^1_{(\geq n\,r\,D),d},\langle M,x\rangle) = (0,\overline{F}_{r,d})\wedge(-1,q^\uparrow_{(\geq n\,r\,D),d})$$

$$\delta_3(q^\downarrow_{(\geq n\,r\,D),d},\langle M,x\rangle) = \bigvee_{n_0+n_1+\ldots+n_k=n}(0,p_{n_0,r,D,d})\wedge\bigwedge_{n_i\neq 0}(i,p_{(\geq n_i\,r\,D),d})$$

$$\delta_3(p_{n,r,D,d},\langle M,x\rangle) = \bigvee_{Y\subseteq M_r(d),|Y|=n}\Big(\bigwedge_{e\in Y}q_{D,e}\wedge\bigwedge_{y\in M_r(d)\setminus Y}q_{\sim D,e}\Big)$$

$$\delta_3(p_{(\geq n\,r\,D),d},\bullet) = \text{if } n=0\text{, then }\ \mathsf{true}\ \text{ else }\mathsf{false}$$

$$\delta_3(p_{(\geq n\,r\,D),d},\langle M,x\rangle) = \begin{cases}\mathsf{false} & \text{if } x\neq r\text{ or } d\text{ not in root cluster}\\ \displaystyle\bigvee_{\substack{n_1+\ldots+n_\ell=n\\ \text{d-respects } M \text{ rel. to } d}}\ \bigwedge_{n_i\neq 0}(0,q^0_{(\geq n_i\,r\,D),a_i}) & \text{otherwise}\end{cases}$$

$$\delta_3(q^{\uparrow}_{(\geq n\, r\, D),d}, \langle M, x\rangle) = (0, q_d) \wedge \big((0, q^0_{(\geq n\, r\, D),d}) \vee (0, q^1_{(\geq n\, r\, D),d})\big)$$

Intuitively, the automaton non-deterministically guesses a partition $n_1 + \ldots + n_k$ of n and verifies that, starting from $\mathbf{a}_i$ at least n_i elements are reachable via r and satisfy D. For each such r-cluster, it proceeds either downwards (in states of the form $q^0_{\cdot}$ and $q^{\downarrow}_{\cdot}$) or looks for the world where the cluster $\mathbf{a}_i$ was a root (in states $q^1_{\cdot}$ and $q^{\uparrow}_{\cdot}$) and proceeds downwards from there on. In states $q^{\downarrow}_{(\geq n\, r\, D),d}$, the automaton again partitions n this time into $n_0, \ldots, n_k$; it then verifies that there are n_0 elements in the r-cluster of d satisfying D and, recursively, that via the i-th successor of the current node, there are n_i elements that are reachable via r and satisfy D. Using the parity condition, we make sure that states $q^{\downarrow}_{(\geq n\, r\, D),d}$ with $n \geq 1$ are not suspended forever, that is, eventualities are finally satisfied.

For the at-most restrictions, recall that $(\leq\, n\; r\; D)$ is equivalent to $\neg(\geq n{+}1\; r\; D)$; we can thus obtain the transitions for $q_{(\leq n\, r\, D),d}$ by "complementing" the transitions for $q_{(\geq n+1\, r\, D),d}$; details are given in the appendix.

In order to construct, for a given query q, an automaton $\mathfrak{A}_q$ which accepts a tree $(T, \tau) \in L(\mathfrak{A}_1)$ iff $\mathcal{I}_{T,\tau} \models q$, we adapt and extend ideas from [8] to canonical tree decompositions. The result is a nondeterministic parity tree automaton [13] of size exponential in q, and doubly exponential in $\mathcal{K}$. In contrast to [8], the query automaton depends on the KB. Indeed, for checking whether a fact $r(x, y)$ from the query is true (given some match candidate), it has to recall domain elements in the states; their number, however, is bounded by the width only. It remains to remark that the question of whether $L(\mathfrak{A}_{\mathcal{K}}) \setminus L(\mathfrak{A}_q)$ is empty can be decided in 3ExpTime, given the mentioned bounds on the sizes of the involved automata.

Theorem 9. *The certain answers problem for P2RPQs over $\mathcal{SQ}$-KBs is decidable in* 3ExpTime.

5 Discussion and Future Work

We have developed a tree-like decomposition for $\mathcal{SQ}$ which handles the interaction of number restrictions over transitive roles and enables the use of automata-based techniques for query answering. Our techniques yield a 3ExpTime upper bound, leaving an exponential gap to the known 2ExpTime lower bound, for answering positive existential queries over $\mathcal{ALC}$ KBs [8].

As immediate future work, we plan to close this gap, taking into account also other techniques for query answering such as rewriting [11]. Another relevant question is the precise *data complexity* – the present techniques give only exponential bounds, but we expect coNP-completeness. Moreover, we plan to extend our approach to nominals and inverses. We will also look at the problem of answering *conjunctive queries (CQs)* in $\mathcal{SQ}$; in general, the proposed automata-based approach yields the same upper bound for the problem of answering P2RPQs or CQs, but we expect it to be easier for CQs. Finally, we plan to see whether our techniques extend to the query containment problem, and develop techniques for *finite* query answering in (extensions of) $\mathcal{SQ}$.

References

1. Amarilli, A., Benedikt, M., Bourhis, P., Vanden Boom, M.: Query answering with transitive and linear-ordered data. In: Proc. of IJCAI-16. pp. 893–899 (2016)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Baader, F., Lutz, C., Sturm, H., Wolter, F.: Fusions of description logics and abstract description systems. J. Artif. Intell. Res. (JAIR) 16, 1–58 (2002)
4. Baget, J., Bienvenu, M., Mugnier, M., Rocher, S.: Combining existential rules and transitivity: Next steps. In: Proc. of IJCAI-15. pp. 2720–2726 (2015)
5. Benedikt, M., Bourhis, P., Vanden Boom, M.: A step up in expressiveness of decidable fixpoint logics. In: Proc. of LICS-16. pp. 817–826 (2016)
6. Bienvenu, M., Ortiz, M., Simkus, M.: Regular path queries in lightweight description logics: Complexity and algorithms. J. Artif. Intell. Res. (JAIR) 53, 315–374 (2015)
7. Calvanese, D., Eiter, T., Ortiz, M.: Regular path queries in expressive description logics with nominals. In: Proc. of IJCAI-09. pp. 714–720 (2009)
8. Calvanese, D., Eiter, T., Ortiz, M.: Answering regular path queries in expressive description logics via alternating tree-automata. Inf. Comput. 237, 12–55 (2014), http://dx.doi.org/10.1016/j.ic.2014.04.002
9. Eiter, T., Lutz, C., Ortiz, M., Simkus, M.: Query answering in description logics with transitive roles. In: Proc. of IJCAI-09. pp. 759–764 (2009)
10. Glimm, B., Horrocks, I., Sattler, U.: Unions of conjunctive queries in SHOQ. In: Proc. of KR-08. pp. 252–262 (2008)
11. Glimm, B., Lutz, C., Horrocks, I., Sattler, U.: Conjunctive query answering for the description logic SHIQ. J. Artif. Intell. Res. (JAIR) 31, 157–204 (2008)
12. Gottlob, G., Pieris, A., Tendera, L.: Querying the guarded fragment with transitivity. In: Proc. of ICALP-13. pp. 287–298 (2013)
13. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games: A Guide to Current Research, LNCS, vol. 2500. Springer
14. Grädel, E., Walukiewicz, I.: Guarded fixed point logic. In: Proc. of LICS-99. pp. 45–54 (1999)
15. Gutiérrez-Basulto, V., Ibáñez-García, Y., Jung, J.C.: Number restrictions on transitive roles in description logics with nominals. In: Proc. of AAAI-17 (2017)
16. Hollunder, B., Baader, F.: Qualifying number restrictions in concept languages. In: Proc. of KR-91. pp. 335–346 (1991)
17. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. Logic Journal of the IGPL 8(3), 239–263 (2000)
18. Kaminski, M., Smolka, G.: Terminating tableaux for $\mathcal{SOQ}$ with number restrictions on transitive roles. In: Proc. of the 6th IFIP TC. pp. 213–228 (2010)
19. Kazakov, Y., Pratt-Hartmann, I.: A note on the complexity of the satisfiability problem for graded modal logics. In: Proc. of LICS-09. pp. 407–416 (2009)
20. Kazakov, Y., Sattler, U., Zolin, E.: How many legs do I have? non-simple roles in number restrictions revisited. In: Proc. of LPAR-07. pp. 303–317 (2007)
21. Pratt-Hartmann, I.: The two-variable fragment with counting and equivalence. Math. Log. Q. 61(6), 474–515 (2015)
22. Rector, A.L., Rogers, J.: Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN. In: In Proc. of RW-06. pp. 197–231 (2006)

23. Schröder, L., Pattinson, D.: How many toes do I have? parthood and number restrictions in description logics. In: Proc. of KR-08. pp. 307–317 (2008)
24. Stevens, R., Aranguren, M.E., Wolstencroft, K., Sattler, U., Drummond, N., Horridge, M., Rector, A.L.: Using OWL to model biological knowledge. International Journal of Man-Machine Studies 65(7), 583–594 (2007)
25. Tendera, L.: Counting in the two variable guarded logic with transitivity. In: Proc. of STACS-05. pp. 83–96 (2005)
26. Vardi, M.Y.: Reasoning about the past with two-way automata. In: Proc. ICALP-98. pp. 628–641 (1998)
27. Wolstencroft, K., Brass, A., Horrocks, I., Lord, P., Sattler, U., Turi, D., Stevens, R.: A little semantic web goes a long way in biology. In: Proc. of ISWC-05 (2005)